

Manual técnico

Objetivos

En el presente documento se pretende dar a conocer de mejor manera cómo es que se ha desarrollado el proyecto de computación gráfica. Así como también detalles importantes dentro del código que deberían ser de utilidad para el usuario.

Diagrama de Gantt

*NOTA: Ver documento html en el repositorio para una mejor vista

Proyecto sin nombre

Inicio: 12 mayo, 2022
Fin: 18 mayo, 2022
Fecha de informe: 12 mayo, 2022

Gráfica Gantt

WBS	Nombre	Trabajo	Semana 20, 2022										Semana 21, 2022					Semana 22, 2022												
			12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8
1	Edificio	2d 4h																												
2	T-Rex	2h																												
2.1	modelado	2h																												
2.2	animación																													
3	Cartel	1h																												
4	Maceta	1h																												
5	Unifila	1h																												
6	Cuadro	1h																												
7	Andamio	1h																												
8	Caja																													
9	Jeep	3h																												
9.1	modelado	2h																												
9.2	animación	1h																												
10	Cisne	1h																												
10.1	modelado	1h																												
10.2	animación																													
11	Animación agua	2h																												
12	Animación puerta																													
13	Reportes	3h																												

Tareas

WBS	Nombre	Inicio	Fin	Trabajo	Terminado
1	Edificio	May 12	May 16	2d 4h	100%
2	T-Rex	May 16	May 16	2h	
2.1	modelado	May 16	May 16	2h	100%
2.2	animación	May 16	May 16		100%
3	Cartel	May 16	May 16	1h	100%
4	Maceta	May 16	May 17	1h	100%
5	Unifila	May 17	May 17	1h	100%
6	Cuadro	May 17	May 17	1h	100%
7	Andamio	May 17	May 17	1h	100%
8	Caja	May 17	May 17		100%
9	Jeep	May 17	May 18	3h	
9.1	modelado	May 17	May 18	2h	100%
9.2	animación	May 17	May 17	1h	100%
10	Cisne	May 18	May 18	1h	
10.1	modelado	May 18	May 18	1h	100%
10.2	animación	May 18	May 18		100%
11	Animación agua	May 18	May 18	2h	100%
12	Animación puerta	May 18	May 18		100%
13	Reportes	May 18	May 18	3h	100%

Alcance del proyecto

El proyecto consta básicamente de dos vistas de un mismo escenario. Para empezar, se seleccionó una temática relacionada con la película *Jurassic Park*. En la primera entrega de esta serie de películas aparece un edificio que funge como edificio donde se recibe a los

visitantes. Así pues, se ha intentado recrear la fachada y la habitación principal del mismo. A continuación se muestran algunos imágenes que han servido de referencia



(Fachada) Centro de visitantes

Se presentan dos imágenes del mismo lugar, sin embargo se propone como principal y referencia de diseño a la segunda. La primera se muestra debido a que la principal no cuenta con los objetos suficientes y se tomarán algunos de la primera.





(Cuarto) Hall Room Centro de visitantes

De estas dos imágenes, se ha recreado el listado siguiente de objetos:

- Maceta con planta.
- Caja de madera.
- Poste unifila.
- Cuadro decorativo.
- Esqueleto T- Rex.
- Letrero de techo.
- Andamio.

Todos estos objetos se encuentran dentro la siguiente ruta relativa del proyecto en un formato *.obj*:

/ProyectoFinal/ProyectoFinal/Models/ObjetosEscenario/

Los archivos se encuentran con sus texturas así como con su archivo *.mtl*, por lo que pueden ser cargados fácilmente en un editor de modelos como lo sería por ejemplo *Maya*. Se insta al usuario a hacer modificaciones en caso de que así quiera hacerlo.

Limitaciones y consideraciones

Hardware y entorno de desarrollo

El proyecto en general ha sido creado y depurado en el entorno de *Visual Studio 2022*, en una máquina con las siguientes características

Ryzen 7 4750U 1700 MHz

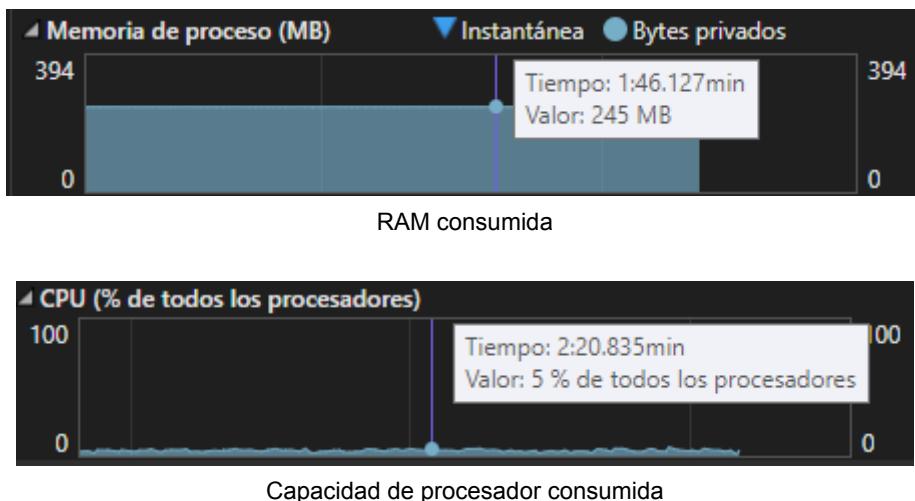
Núcleos 8

Hilos 16

GPU Integrada AMD Radeon RX Vega 7 1600MHz

8 GB RAM SDIM DDR4 3200MHz

Notándose que el rendimiento gráfico es bueno, por lo que se expone como punto de referencia a los recursos de cada usuario y su consideración a la hora de ejecutar el proyecto. A continuación se muestran los recursos de RAM y capacidad del procesador que consume el programa en plena ejecución:



Así pues, se recomienda tener instalado una versión de OpenGL 2.0 o superior.

Documentación

Archivos de cabecera

En el proyecto se usaron varias bibliotecas que también se encuentran dentro del proyecto, describiendo las más importantes a continuación:

Header	Descripción
GL/glew.h	(OpenGL Extension Wrangler-Library) Biblioteca que carga punteros a OpenGL en tiempo de ejecución. Es requerida para acceder a ciertas funciones de OpenGL a partir de la versión 1.1.
GLFW/glfw3.h	(Graphics Library Framework) Biblioteca para uso con OpenGL para crear y dirigir ventanas así como entrada de joystick, teclado y ratón
SOIL2/SOIL2.h	Biblioteca usada para la carga de texturas
Shader.h	Biblioteca encargada de cargar shaders

	creados por el usuario. Se precisa tener el <i>Vertex Shader</i> y <i>Fragment Shader</i> , pues ambos archivos se compilan en tiempo de ejecución para añadirse al programa
Camera.h	Se encarga de hacer el procesamiento de la cámara que se usa para visualizar el entorno.
Model.h	Biblioteca encargada de hacer la carga de modelos 3D en un formato <i>.obj</i> . Por lo mismo, se recomienda tener en la misma ruta de carga la imagen de textura en un formato PNG y tamaño 512x512 píxeles.
Texture.h	Archivo de cabecera encargado de la carga de textura cuando esta no es parte de un modelo en particular.

Funciones del programa

Función	Descripción
KeyCallback	Hace función de escucha para detectar eventos del teclado. Aquí se disparan acciones cuando estos eventos suceden.
MouseCallback	Función que hace de escucha para eventos en el ratón. Su función principal es controlar la cámara.
DoMovement	Se encarga de administrar el movimiento de la cámara a través del teclado. También de controlar la posición del cubo que sirve para medir posiciones.
animacion	Función encargada de todas las acciones concernientes a las animaciones del escenario
main	Función principal donde se realizan todas las configuraciones pertinentes del programa y donde estará el while que se encargará de refrescar la pantalla constantemente.

Shaders utilizados

Shader	Descripción
lightingShader	Procesa los modelos para agregar la interacción de la luz con estas. Ya sea con

	la luz ambiental, especular o de lámpara. También permite activar el canal alpha para procesar la transparencia de algunos objetos.
animAguaShader	Shader se encarga de animar una textura como el agua al manipular sus vértices y textura para generar un movimiento.

Variables de interés

Variable	Descripción
camera	Guarda la posición de la cámara en el escenario
carSpeed	controla la velocidad de desplazamiento del Jeep
iniPosCar	Vector con la posición inicial del carro
swanSpeed	Controla la velocidad de desplazamiento del cisne
iniPosSwan	Vector con la posición inicial del cisne
dinoSpeed	Controla la velocidad del baile del dinosaurio

*NOTA: Al realizar un cambio en la posición inicial de alguno de los objetos animados debe considerarse de nuevo los límites para la animación definidos dentro de la función *animacion()*.

Conclusiones

- Al principio del semestre me parecía sumamente ambicioso el imaginar hacer una animación casi desde 0. Al terminar el proyecto se entiende que todo lo grande parte de cosas sencillas para hacerlas grandes.
- Una herramienta gráfica para la edición de modelos facilita en desmesura las actividades del modelado. Si no se hubiera contado con una, no se hubiera podido realizar los avances que se hicieron.
- En maya se visualizó que hay herramientas para la animación. Esto permite pensar que al igual que el modelado, la animación se debe facilitar mucho más haciendo uso de un programa como Maya.

Technical manual

Objectives

This document presents the development way for the project of graphic computer subject and some technical considerations. Also, it contains the most important things in the source code that users should know.

Gantt Diagram

**NOTE: Inside the repository are a html document which have a better view of the gantt diagrams*

Tareas

WBS	Nombre	Inicio	Fin	Trabajo	Terminado
1	Edificio	May 12	May 16	2d 4h	100%
2	T-Rex	May 16	May 16	2h	
2.1	modelado	May 16	May 16	2h	100%
2.2	animación	May 16	May 16		100%
3	Cartel	May 16	May 16	1h	100%
4	Maceta	May 16	May 17	1h	100%
5	Unifila	May 17	May 17	1h	100%
6	Cuadro	May 17	May 17	1h	100%
7	Andamio	May 17	May 17	1h	100%
8	Caja	May 17	May 17		100%
9	Jeep	May 17	May 18	3h	
9.1	modelado	May 17	May 18	2h	100%
9.2	animación	May 17	May 17	1h	100%
10	Cisne	May 18	May 18	1h	
10.1	modelado	May 18	May 18	1h	100%
10.2	animación	May 18	May 18		100%
11	Animación agua	May 18	May 18	2h	100%
12	Animación puerta	May 18	May 18		100%
13	Reportes	May 18	May 18	3h	100%

Project Scope

This project is based on the Jurassic Park movie, so in some scenes of this movie a building appears which has a visitor receiver function, probably in the first movie . The front of this building has been selected to be recreated and the main room too. The following images were the image references.



(Building front) Visitor Center

There are two images of the same place because the second image hasn't enough stuff to be recreated. To be clear, I take some objects from both images but the second is the main which project has been based on.



(Main Room) Hall Room

The recreated objects are:

- Flower pot.
- Transport box
- Barrier
- Ornamental picture
- T-Rex skeleton
- Poster
- Scaffold

All of them are contained on the following path with a *.obj* format:

/ProyectoFinal/ProyectoFinal/Models/ObjetosEscenario/

Texture and *.mtl* files are in the same folder too and you should be free to easily modify the recreated objects using a model editor like *Maya* or whatever you want. Feel free to do it.

Limitation and consideration Limitaciones y consideraciones

Hardware and development environment

All the project has developed and debugged in *Visual Studio 2022*, using a computer with the next characteristics:

Ryzen 7 4750U 1700 MHz

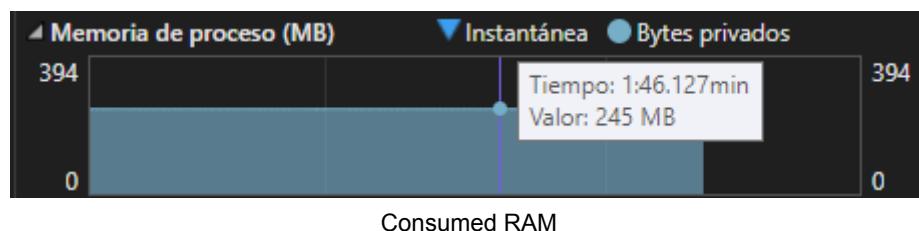
Cores 8

Threads 16

Integrated GPU AMD Radeon RX Vega 7 1600MHz

8 GB RAM SDIM DDR4 3200MHz

During the program running I noticed a good performance, so you can take my computer characteristics like a comparative point if you try to execute the program on another machine. The following pictures show the RAM and processor capacity consumed during the program execution.



Also, it is highly recommended to have OpenGL 2.0 or a higher version.

Documentation

Header Files

This project use a variety of libraries which also are in the repository, but I describe ones of the important at next:

Header	Descripción
GL/glew.h	(OpenGL Extension Wrangler-Library) This library is required to use some

	functions in higher versions than OpenGL 1.1
GLFW/glfw3.h	(Graphics Library Framework) This library is required to use some window, keyboard and mouse functions with OpenGL.
SOIL2/SOIL2.h	The main function is to charge textures.
Shader.h	This library is required to load the user's shaders. Is strictly recommended to have the <i>Vertex Shader and Fragment Shader files</i> 'cause these are compiled and integrated to the program in execution time.
Camera.h	This header file has the function to administrate camera functions.
Model.h	This library charges the 3D models with a .obj format. Is important to have in the same folder the texture image with a PNG format with a 512x512 size and the .mtl file.
Texture.h	The main function is to charge textures which do not have a specific model into the programme.

Programe functions

Función	Descripción
KeyCallback	This function manages events which happen mainly at the keyboard. Actions are taken when events happen.
MouseCallback	Function which manages mouse events. Always to control the camera rotation.
DoMovement	Function which manages mouse events. Usually to control the camera movement.
animacion	This function manages all the animations that happen during the program execution.
main	Function which configures some start values and refreshes the window image using a while function.

Used Shaders

Shader	Descripción

lightingShader	This shader can process a model with a light that interacts with it. Also, this shader can activate the alpha channel which makes the transparency possible.
animAguaShader	This shader can animate the water effect, so is their only function in the program.

Important variables

Variable	Descripción
camera	It is a vector which has the initial position of the camera.
carSpeed	Controls the jeep speed.
iniPosCar	It is a vector which has the initial position of the jeep.
swanSpeed	Controls the swan speed.
iniPosSwan	It is a vector which has the initial position of the swan.
dinoSpeed	Controls the dancing speed of the T-Rex skeleton.

*NOTE: If you change the initial position value of any object which is animated, is important reconfigure the behavior defined at the **animacion** function.

Conclusions

- At the beginning of the course I thought it wasn't possible to do a complete animation only with this time. In the end I understand that all big things start with small and easy concepts.
- Graphic tools for model edition are very useful to decrease modeling time. I don't want to imagine what would happen if tools like Maya didn't exist.
- I saw Maya has some animation tools. So, I think animation can be easily done in a graphic environment too.